NXP Semiconductors
Errata

Document identifier: IMX8MDQLQ_2N14W
Rev. 3, 6/2024

IMX8MDQLQ_2N14W

Mask Set Errata

Mask Set Errata for Mask 2N14W

Revision History

This report applies to mask 2N14W for these products:

- PIMX8MQ6DVAJZAB
- MIMX8MQ6DVAJZIB
- MIMX8MQ6DVAJZAB
- PIMX8MQ6CVAHZAB
- MIMX8MQ5DVAJZAB
- MIMX8MQ5CVAHZAB
- MIMX8MD6DVAJZAB
- MIMX8MQ6CVAHZAB
- MIMX8MD6CVAHZAB

Table 1. Revision History

Revision	Date	Significant Changes
3	6/2024	The following errata were revised. • ERR051126
2	1/2023	The following errata were added. • ERR051273 • ERR051249
1	3/2022	The following errata were added. • ERR051126 • ERR050805 • ERR006941 • ERR050384 The following errata were revised. • ERR009165 • ERR003774 • ERR006940
0	10/2018	Initial Revision

Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR003774	AIPS: Unaligned access to AIPS internal registers will result in an abort response.

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR011174	CA53: Cannot enter WAIT mode
ERR011171	CA53: Cannot support single-core runtime wakeup
ERR006941	Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/ JTAG Debug Port
ERR006939	Core: Interrupted loads to SP can cause erroneous behavior
ERR009004	Core: ITM can deadlock when global timestamping is enabled
ERR009005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
ERR006940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
ERR011167	CoreSight: No timestamp in trace data
ERR011039	DCSS: AYUV to YUV422 issue in Scaler
ERR010951	DCSS: CTX_LD does not support SB_COUNT.HP_COUNT = 0 if SB_COUNT.LP_COUNT != 0.
ERR011041	DCSS: DPR-DTRC integration limitation when DTRC is used for decompression/de-tiling
ERR011042	DCSS: DTRC requires 16 lines when enabled
ERR010911	DDR Perf Monitor: Counter0 of DDR performance monitor cannot be overridden
ERR011326	DDRC: ARQOS and AWQOS are fixed to 0 on DRAM controller AXI port
ERR051273	DDRC: Periodic hardware-based DQS2DQ calibration is not supported
ERR050805	DRAM: Controller automatic derating logic may not work when the LPDDR4 memory temperature is above 85ºC at initialization
ERR009535	ECSPI: Burst completion by SS signal in slave mode is not functional
ERR009165	ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice
ERR007805	I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min
ERR050384	MIPI CSI: Receive FIFO Overflow may lead to system hang
ERR011193	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires
ERR010740	QuadSPI: Insufficient read data may be received in the RX Data Buffer register
ERR008565	QuadSPI: Low performance received when using certain access sequences
ERR011096	SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1
ERR011150	SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0
ERR051249	TZASC: ID SWAP function not supported
ERR051126	USB3: Unreliable receiver detection in low power P3 mode at all temperatures
ERR011324	USB3: USB device fails to connect after system resumes from DSM
ERR011231	USB: Clock must remain on during suspend/resume
ERR011176	USB: USB 3.0 Host Hot Plug incorrectly enumerates

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR011232	USDHC: uSDHC setting requirement for IPG_CLK and AHB_BUS clocks

Known Errata

ERR003774: AIPS: Unaligned access to AIPS internal registers will result in an abort response.

Description

Unaligned access to AIPS internal registers will return an abort response.

Workaround

Only aligned AIPS internal register access is supported. Software should not issue unaligned accesses to AIPS internal registers.

ERR011174: CA53: Cannot enter WAIT mode

Description

CA53 platform cannot enter WAIT mode if there is a pending interrupt. If the chip enters WAIT (WFI) mode, the chip requires a reboot to recover.

Workaround

No workarounds. SW should not use WAIT mode.

Impact: This mode turns off the power to the SCU (Snoop Control Unit) and the L2 cache. Not having this mode affects only 1 mode of core power savings.

ERR011171: CA53: Cannot support single-core runtime wakeup

Description

According to the GIC500 specification and the Arm Trusted Firmware design, when a CPU core enters the deepest CPU idle state (power-down), it must disable the GIC500 CPU interface and set the Redistributor register to indicate that this CPU is in sleep state. In such case, if the CPU core is in WFI or power-down with CPU interface disabled, another core cannot wake-up the powered-down core using SGI interrupt.

Workaround

One workaround is to use another A53 core for the IRQ0 which is controlled by the IOMUX GPR to generate an external interrupt to wake-up the powered-down core.

The SW workaround is implemented into default BSP release. The workaround commit tag is “MLK-16804-04 driver: irqchip: Add IPI SW workaround for imx8mq".

ERR006941: Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/JTAG Debug Port

Description

Arm Errata 771919: Asynchronous sampling of SWDIOTMS might cause incorrect operation of

SerialWire/JTAG Debug Port

Status

Affects: Cortex-M4, Cortex-M4F

Fault Type: Implementation Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

Description

The signal SWDIOTMS is bi-directional and can be driven from either the debugger or the SWJ-DP, or pulled up by an external resistor during the turnaround periods.

The SerialWire protocol is defined with a high PARK bit at the end of the header before the turnaround period that precedes the ACK from the SWJ-DP. This ensures that the line is high, and the resistor keeps it high during the ACK period. Therefore, if the SWJ-DP does not respond, the debugger will reliably sample the line SWDIOTMS high during the missing ACK.

However, during the turnaround period after the ACK or read data there is no PARK bit to guarantee that the line is high immediately before the turnaround period. In this case, if the pull-up resistor does not pull the line high within a single SWCLKTCK cycle, the incorrect state of SWDIOTMS might be sampled.

Functionally, the logic is insensitive to the state of SWDIOTMS during these periods, but synthesis tools might introduce multiple path logic that is sensitive to SWDIOTMS glitches around the clock edges.

Conditions

All write transactions and some read transactions might be vulnerable to this erratum when both:

- Serial Wire mode is being used
- The physical implementation does not prevent glitch generation.

Implications

The SWJ-DP might sample SWDIOTMS incorrectly and enter an UNPREDICTABLE state. At the time of publication, ARM is not aware of any reports of observed failures due to this erratum.

Workaround

Check the following points after implementation:

1) Ensure that the evaluation of NextState in DAPSwjWatcher.v is not sensitive to SWDITMSSync1 when State_cdc_check has the value 10'b1100100000 (SWJ_SSLP).

2) Ensure that the following logic in DAPSwDpProtocol.v is implemented using AND gates or a CDC-safe mux for each bit:

assign ResetCountD = DBGDI & ~DBGDOEN ? (ResetCountReg+6'd1) : {6{1'b0}};

3) Ensure that the ResetCountReg flops in DAPSwDpProtocol.v are implemented using metastability-hardened cells if possible.

4) Ensure that the evaluation of NxtState in DAPSwDpProtocol.v is insensitive to DBGDI when State has any of the following values:

- 5'b01000 (SWDP_SLEPARKH)
- 5'b01010 (SWDP_SLETRNH2)
- 5'b01011 (SWDP_SLETRNH1)
- 5'b01100 (SWDP_SLETRNH0)
- 5'b10011 (SWDP_SLEPARKW)

- 5'b10100 (SWDP_SLETRNW3)

- 5'b10101 (SWDP_SLETRNW2)

- 5'b10110 (SWDP_SLETRNW1)

- 5'b10111 (SWDP_SLETRNW0)

5) Ensure that the following flops in DAPSwDpProtocol.v are implemented with CDC-safe recirculation muxes:

- SerBank

- SerDir

- SerAddr

- ShiftReg

- Parity

- ErrorChk

- WriteErr

- WbufReq

6) Ensure that the following flops in DAPJtagDpProtocol are implemented with CDC-safe recirculation muxes:

- JTAGcurr

ERR006939: Core: Interrupted loads to SP can cause erroneous behavior

Description

Arm Errata 752770: Interrupted loads to SP can cause erroneous behavior

This issue is more prevalent for user code written to manipulate the stack. Most compilers will not be affected by this, but please confirm this with your compiler vendor. MQX™ and FreeRTOS™ are not affected by this issue.

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

3) LDR SP,[Rn,#imm]

4) LDR SP,[Rn]

5) LDR SP,[Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

Conditions:

1) An LDR is executed, with SP/R13 as the destination.

2) The address for the LDR is successfully issued to the memory system.

3) An interrupt is taken before the data has been returned and written to the stack-pointer.

Implications:

Unless the load is being performed to Device or Strongly-Ordered memory, there should be no implications from the repetition of the load. In the unlikely event that the load is being performed to Device or Strongly-Ordered memory, the repeated read can result in the final stack-pointer value being different than had only a single load been performed.

Interruption of the two write-back forms of the instruction can result in both the base register value and final stack-pointer value being incorrect. This can result in apparent stack corruption and subsequent unintended modification of memory.

Workaround

Most compilers are not affected by this, so a workaround is not required.

However, for hand-written assembly code to manipulate the stack, both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

If repeated reads are acceptable, then the base-update issue may be worked around by performing the stack pointer load without the base increment followed by a subsequent ADD or SUB instruction to perform the appropriate update to the base register.

ERR009004: Core: ITM can deadlock when global timestamping is enabled

Description

ARM ERRATA 806422

The Cortex-M4 processor contains an optional Instrumentation Trace Macrocell (ITM). This can be used to generate trace data under software control, and is also used with the Data Watchpoint and Trace (DWT) module which generates event driven trace. The processor supports global timestamping. This allows count values from a system-wide counter to be included in the trace stream.

When connected directly to a CoreSight funnel (or other component which holds ATREADY low in the idle state), the ITM will stop presenting trace data to the ATB bus after generating a timestamp packet. In this condition, the ITM_TCR.BUSY register will indicate BUSY.

Once this condition occurs, a reset of the Cortex-M4 is necessary before new trace data can be generated by the ITM.

Timestamp packets which require a 5 byte GTS1 packet, or a GTS2 packet do not trigger this erratum. This generally only applies to the first timestamp which is generated.

Devices which use the Cortex-M optimized TPIU (CoreSight ID register values 0x923 and 0x9A1) are not affected by this erratum.

Workaround

There is no software workaround for this erratum. If the device being used is susceptible to this erratum, you must not enable global timestamping.

ERR009005: Core: Store immediate overlapping exception return operation might vector to incorrect interrupt

Description

Arm Errata 838869: Store immediate overlapping exception return operation might vector to incorrect interrupt

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

The Cortex-M4 includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

Configurations Affected

This erratum only affects systems where writeable memory locations can exhibit more than one wait state.

Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

```
...
schedule_barrier();
asm{DSB};
schedule_barrier();
}
```

GCC:

```
...
asm volatile ("dsb 0xf" ::: "memory"); volatile ("dsb 0xf" ::: "memory");
}
```

ERR006940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Description

Arm Errata 776924: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Affects: Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

On Cortex-M4 with FPU, the VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

Workaround

A workaround is only required if the floating point unit is present and enabled. A workaround is not required if the memory system inserts one or more wait states to every stack transaction.

There are two workarounds:

1) Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).

2) Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

ERR011167: CoreSight: No timestamp in trace data

Description

During the JTAG debug with trace function, there is no timestamp in the trace data. Because of the lack of a timestamp, the debug with trace function is not enabled by CoreSight.

Workaround

No workarounds, SoC does not support this feature.

ERR011039: DCSS: AYUV to YUV422 issue in Scaler

Description

Scaler uses 5 tap filtering for Graphics pipe. When the input data format in the graphics pipe is AYUV444 and dolby path is used, scaler is programmed to output YUV422 and src_is_video is set to 1. In this case, the module fails to replicate the last row when the vertical tap filter crosses the bottom boundary of the source picture. It results in incorrect pixels being output for the last 2 rows of the image.

Workaround

When the Input image format is AYUV444 and the Dolby Path is used, i.e. scaler converting AYUV444 to YUV422, program the src_is_video to 1, and set the Source Picture Height Register field to (input_image_height - 2).

No artifacts are found by this work around. The change is to be done only to the register programming, i.e. the input height to be used for determining the scaling coefficients should remain the same.

If the input image height is 256, and the Scaler is configured to convert AYUV444 to YUV422, source picture height register should be programmed to 254,

in all the other cases, it should be programmed to 255.

ERR010951: DCSS: CTX_LD does not support SB_COUNT.HP_COUNT = 0 if SB_COUNT.LP_COUNT != 0.

Description

The CTX_LD does not support single buffer high priority (SB.HP) count of 0 if the single buffer low priority (SB.LP) count is not 0. So, if any single buffered registers need to be programmed, the SB.HP count must be > 0.

Workaround

SW must always program at least one entry for the SB.HP region when there are entries in the SB.LP region. If the SB.LP region is 0, it is ok for both regions to be =0.

ERR011041: DCSS: DPR-DTRC integration limitation when DTRC is used for decompression/de-tiling

Description

The DTRC does not allow re-read of pixels. DTRC requires DPR to read exactly full scan line pixels when DTRC is used for decompression/de-tiling. The AXI interface on DTRC is a 128-bit which requires picture size x bitdepth/8 to be an integer multiple of 16. On further restriction, DPR requires X-WIDE to be integer multiple of 64bytes, which does not allow any picture size.

Workaround

• DPR requires all transactions to be 64B, which forces 10b buffers to be constrained to (or padded to) a multiple of 256 pixels wide.

• There are a some hardware constraints on how the blocks access the buffers:

• There is no stride in G2/G1 compressed/uncompressed tiled output, so we cannot pad dummy data directly in those formats.

• G2/G1 raster scan output also does not support stride, so we cannot bypass DTRC to directly read raster scan output from G2 to solve the issue.

• DTRC raster scan output is consistent with the G2 raster scan output and, consequently, also does not support stride.

SW Work around:

• The DTRC crop feature can be used to simulate 'padding' in some instances. By setting the parameters to a negative crop value on the right edge, the DTRC will respond by padding with some limitations.

• This ‘padding’ will always work for 8-bit uncompressed G1/G2 output, and 10-bit uncompressed G2 output.

• This ‘padding’ will only work for 8-bit/10-bit compressed G2 output on cases that the padded picture width is <= (128 * ceiling(original_picture_width_in_pixels / 128))

Notes:

Almost all 8b compressed surfaces should work fine due to the ability to pad up to 128 pixel multiples. For 10b compressed surfaces that this does not work with (e.g. 1080p), NXP recommends that the reference frame compression be disabled on G2. The uncompressed tile format should be readable and possible to pad on the DTRC.

That said, NXP confirmed that only 4K formats should be requiring 10b (all of which should work compressed). This means that all primary use cases should be possible to support with compressed buffers.

Programming Model to use Padding feature (highlighted only registers which requires or have programming difference)

DTRC programming:

F0SIZE - The picture size still needs to be set to the original picture size.

F0CROPORIG - Cropped picture top left origin is set to (0, 0).

F0CROPSIZE - For 8bit crop picture width should be (orig picture width + 63) & 0xffff_ffC0; For 10bit crop picture width should be (orig picture width + 255) 0xffff_ff00 (make picture width 256 pixels aligned)

F0DCTL - Set bit 18 to 1'b, Set bit 17 to 0 for compressed source frame buffer and 1 for uncompressed source frame buffer.

DPR programming -

FRAME CTRL0[PITCH] = Cropped width (programmed in DTRC F0CROPSIZE reg) * bit_depth/8

FRAME_1P_PIX_X_CTRL[NUM_X_PIX_WIDE] = Cropped width (programmed in DTRC F0CROPSIZE reg) * bit_depth/8

FRAME_2P_PIX_X_CTRL[NUM_X_PIX_WIDE] = Cropped width (programmed in DTRC F0CROPSIZE reg) * bit_depth/8

Scaler shall be programmed with Original picture size.

ERR011042: DCSS: DTRC requires 16 lines when enabled

Description

On video channels, the DPR requires both the chroma and luma height to be multiples of 8. If not a multiple of 8, it can cause chroma height to be different than half the luma height.

Because there are not separate DTRC registers for the luma and chroma height and there is only one value, the chroma will always be exactly half the luma.

When the chroma height or luma height does not match between the DTRC and DPR, the DPR will request data that the DTRC doesn't have, therefore, the DTRC will not respond to the request. This will cause an incomplete transaction and stall the bus after enough transactions occur.

Example: For a 360 line image. DTRC = 360 lines, DPR Luma = 360, DPR chroma = 184 (it must be a multiple of 8). Because the DPR chroma and DTRC lines/2 are not the same, this issue will arise.

Workaround

Bypassing the DTRC will not cause this issue to arise. Padding the frame to be a multiple of 16 will work as well.

ERR010911: DDR Perf Monitor: Counter0 of DDR performance monitor cannot be overridden

Description

Counter0 will generate interrupts only when full. The Counter Parameter (CP) field cannot be written to, limiting the performance monitoring period to the full 32-bit count value.

Workaround

No workaround. This function is only for development debug purpose.

ERR011326: DDRC: ARQOS and AWQOS are fixed to 0 on DRAM controller AXI port

Description

QoS priority settings are fixed and non-programmable in the DDR controller. Therefore, all accesses to the DRAM controller are fixed to the same priority. This limits the flexibility of the DDR QoS function.

Typical symptoms of this errata include flickering of the display, and overruns of MIPI-CSI interface.

Workaround

There are system level settings such as using the NOC and the DDR PHY to limit accesses.

As an example work-around, please see the HoC/DDR changes in IMX8MM GA released BSP, changes that were made in linux-imx,imx-atf, and uboot-imx NXP git repository.

ERR051273: DDRC: Periodic hardware-based DQS2DQ calibration is not supported

Description

If periodic hardware-based DQS2DQ calibration is enabled, a potential timing condition in the DDR controller interface logic can cause a random hang or lock-up.

Workaround

Disable the automatic DQS2DQ periodic training by disabling the DFI PHY Master interface logic from both the DDR controller and PHY.

DDR Controller: Change the RPA excel file > “Register configuration” sheet > “DFI_PHYMSTR_EN” from a value of 1 to 0.

PHYMSTR.dfi_phymstr_en = 0x0

DDR PHY: set the following PHY parameters to disable the PHY Master interface logic

ddrphy_reg(0x20010)= 0x0

An updated DDR tool will be made available that supports this PHY configuration.

Currently, DQS2DQ calibration only takes place on power-up and when resuming from low power modes. To date, no failures or stability issues have been observed by disabling this hardware-based DQS2DQ periodic training.

Please contact your NXP representative or submit a support ticket for additional details if required.

ERR050805: DRAM: Controller automatic derating logic may not work when the LPDDR4 memory temperature is above 85ºC at initialization

Description

LPDDR4 memories require periodic refreshes to maintain memory contents. Per the JEDEC specification JESD209-4 the memory refresh rate needs to increase and timings de-rated as the memory operational temperature exceeds vendor-defined temperature thresholds. The LPDDR4 Mode Register 4 (MR4) contains temperature/refresh rate information and a Temperature Update Flag (TUF).

An issue exists with the automatic derating logic of the DDR controller that only samples the LPDDR4 MR4 register when the Temperature Update Flag (TUF) field (MR4[7]) is 1’b1. If the LPDDR4 memory is initialized and starts operation above 85ºC (MR4[2:0] > 3’b011), the MR4 Temperature Update Flag (TUF) will not be set. The DDR Controller will therefore not automatically adjust the memory refresh rate or de-rate memory timings based on the LPDDR4 memory temperature. This may cause the controller incorrectly setting the refresh period, potentially cause the LPDDR4 memory losing data contents and lead to possible data integrity issues above 85ºC. The actual memory temperature threshold values may vary depending on memory vendors.

If the LPDDR4 memory temperature remains below 85ºC at initialization (Consumer-grade memory devices), then the derating logic works as intended, automatically adjusting the memory refresh period and memory timing during the entire system operation. The issue does not occur in this specific scenario since derating is not required.

This erratum does not impact other SoC supported DDR memory interfaces such as DDR4 or DDR3L.

Workaround

Analysis must be performed to determine if the particular design use case could be affected by the errata. There are three options to choose from based on this assessment:

Option 1: Keep the automatic derating logic of the DDR controller enabled. This is the default option and is suitable for designs that are not designated to boot at or above 85° (designs not affected by the errata).

Option 2: Disable the automatic derating logic of the DDR controller and apply fixed x2 refresh rate (0.5x refresh). This option is suitable for designs that are expected to boot at or above 85°C and memory’s MR4[2:0] (Refresh Rate) DOES NOT report the following conditions:

3b101: 0.25x refresh, no de-rating

3b110: 0.25x refresh, with de-rating

3b111: SDRAM High temperature operating limit exceeded

Option 3: Disable the automatic derating logic of the DDR controller and apply fixed x4 refresh rate (0.25x refresh) together with derated timings for tRCD, tRAS, tRC, tRP and tRRD. This option is suitable for designs that are expected to boot at or above 85°C and memory’s MR4[2:0] (Refresh Rate) DOES report the following conditions:

3b101: 0.25x refresh, no de-rating

3b110: 0.25x refresh, with de-rating

3b111: SDRAM High temperature operating limit exceeded

All the three options are implemented in the Register Programming Aid (RPA) spreadsheet (version v15 and later) where users can select the option that fits their use case.

For Option 2 and Option 3 it is highly recommended to consult with the DRAM vendor on supported temperature grades.

ERR009535: ECSPI: Burst completion by SS signal in slave mode is not functional

Description

According to the eCSPI specifications, when eCSPI is set to operate in the Slave mode (CHANNEL_MODE[x] = 0), the SS_CTL[x] bit controls the behavior of burst completion.

In the Slave mode, the SS_CTL bit should control the behavior of SPI burst completion as follows:

- 0—SPI burst completed when (BURST_LENGTH + 1) bits are received
- 1—SPI burst completed when the SS input is negated

Also, in BURST_LENGTH definition, it is stated “In the Slave mode, this field takes effect in SPI transfer only when SS_CTL is cleared.”

However, the mode SS_CTL[x] = 1 is not functional in Slave mode. Currently, BURST_LENGTH always defines the burst length.

According to the SPI protocol, negation of SSB always causes completion of the burst. However, due to the above issue, the data is not sampled correctly in RxFIFO when {BURST_LENGTH+1}mod32 is not equal to {actual burst length}mod32.

Therefore, setting the BURST_LENGTH parameter to a value greater than the actual burst does not resolve the issue.

Workaround

Do not use the SS_CTL[x] = 1 option in the Slave mode. The accurate burst length should always be specified using the BURST_LENGTH parameter.

ERR009165: ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice

Description

When using DMA to transfer data to the TXFIFO, if the data is written to the TXFIFO during an active ECSPI data exchange, this can cause a glitch in the TXFIFO empty signal, resulting in the TXFIFO read pointer (TXCNT) not updating correctly, which in turn results in the current transfer getting resent a second time.

Workaround

This errata is only seen when the SMC (Start Mode Control) bit is set. A modified SDMA script with TX_THRESHOLD = 0 and using only the XCH (SPI Exchange) bit to initiate transfers prevents this errata from occurring. There is an associated performance impact with this workaround. Testing transfers to a SPI-NOR flash showed approximately a 5% drop in write data rates and a 25% drop in read data rates.

ERR007805: I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min

Description

When the I2C module is programmed to operate at the maximum clock speed of 400 kHz (as defined by the I2C spec), the SCL clock low period violates the I2C spec of 1.3 uS min. The user must reduce the clock speed to obtain the SCL low time to meet the 1.3us I2C minimum required. This behavior means the SoC is not compliant to the I2C spec at 400kHz.

Workaround

To meet the clock low period requirement in fast speed mode, SCL must be configured to 384KHz or less.

ERR050384: MIPI CSI: Receive FIFO Overflow may lead to system hang

Description

When receive FIFO data in MIPI CSI can not be written to memory in time if the system is heavily loaded with DDR memory requests, overflow will happen. It may lead to illegal AHB bus access at CSI AHB master port and then causes system hang.

Note that the possibility of MIPI CSI1 overflow will be lower than MIPI CSI2. MIPI CSI1 can support 4 outstanding write transition, however MIPI CSI2 can support 2 outstanding write transition. So MIPI CSI1 will realize more memory bandwidth.

Any use case can create the CSI FIFO to overflow when multiple master arbitrate for system DRAM bandwidth.

One example scenario, which might trigger the erratum, is described below:

- 2 x 1080p display output (Display refresh being a real time process consuming BW over time).

- 1 x 1080p50 camera in via MIPI CSI1 or CSI2 (Note, overflow is more likely on CSI2 due to fewer outstanding write transaction support limitations)

Workaround

The recommended workaround for this erratum is to avoid MIPI CSI overflow happen. Two undocumented debug registers can be used to monitor CSI FIFO levels in real time. When FIFO reaches one threshold, disable CSI and restart CSI.

For example, create one program on M4 core to monitor CSI FIFO level and do as follows:

1) Read the CSI FIFO debug registers: Register CSI1_CSICR19 (0x30A9_004C) for MIPI CS1 or register CSI2_CSICR19 (0x30B8_004C) for MIPI CSI2

2) If the value of register CSIx_CSICR19 is larger than 192, Set register CSI2RX_CFG_DISABLE_DATA_LANES (0x30A7_0104 for MIPI CS1 or 0x30B6_0104 for MIPI CSI2) as 0xFF to de-assert MIPI CSI enable signals.

3) Wait for 3us. Jump to step 1).

ERR011193: PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires

Description

Impacted Configuration(s): Upstream Port configurations:

CC_DEVICE_TYPE =DM and device_type =4'b0000

Defect Summary:

When a function issues a PM_PME Message, it sets the PME_Status bit. If the Downstream port has not cleared the PME_Status bit within 100ms, a PME Service timeout occurs. At this point, the Upstream port must resend the PM_PME message.

In the current implementation of the controller, the PME Service timeout does not trigger an exit from L1 to resend the PM_PME message.

System Usage Scenario:

Upstream ports using a wake-up mechanism followed by a power management event (PME) message.

Consequence(s):

The defect has the following effect:

The PME service routine cannot make forward progress until the PM_PME message is resent.

Workaround

Poll the PME_Status bit after sending the PME message to exit L1 state. If this bit remains 1 for 100ms or more, SW must re-toggle bit 10 "PCIE_CTRL_APPS_PME" of register "SRC_PCIEPHY_RCR" .

ERR010740: QuadSPI: Insufficient read data may be received in the RX Data Buffer register

Description

Data read from flash through QuadSPI using Peripheral Bus Interface (IPS) may return insufficient data in the RX Buffer Data register (QuadSPI_RBDRn) when the read data size of a flash transaction is programmed to be greater than 32 bytes.

Workaround

For data size greater than 32 bytes, program the IP data transfer size in the IP configuration register (QuadSPI_IPCR[IDATSZ]) to be in multiples of 8 bytes.

ERR008565: QuadSPI: Low performance received when using certain access sequences

Description

When sequential accesses are made to QuadSPI with prefetch enabled and the memory difference between two subsequent access bursts is very small then the performance observed may vary with different flash frequency. As an example this may occur if a pre-fetch is requested when an ongoing burst would already fetch the required data.

The reduced performance can occur when using the DMA or cores and when performing execute-in-place in particular.

No data corruption is observed but effective performance can be reduced by more than half.

Workaround

The impact can be limited by reducing the number of overlapping QuadSPI requests through optimizing the size of the pre-fetch buffer but cannot be completely eliminated.

ERR011096: SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1

Description

When the SAI transmitter or receiver is configured for internal bit clock with BCI = 1, the bit clock is not generated for either of the following two configurations:

a) SYNC = 00 and BCS = 0

b) SYNC = 01 and BCS = 1

Workaround

When the SAI transmitter or receiver is configured for internal bit clock with BCI=1, use only one of the following two configurations:

a) SYNC = 01 and BCS = 0

b) SYNC = 00 and BCS = 1

ERR011150: SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0

Description

If the receive or transmit bit clock (BCLK) is internally generated, enabled with DIV > 0 and is then disabled, due to software or Stop mode entry, and the BCLK is enabled again, the clock is not generated.

Workaround

If the receive or transmit BCLK is internally generated and a DIV value greater than 0 is used, the SAI must be reset before the BCLK is re-enabled. This is achieved by writing the SR bit in the respective RCSR or TCSR register first to 1 and then immediately to 0.

ERR051249: TZASC: ID SWAP function not supported

Description

The TZASC ID SWAP function may un-expectantly grant non-secure access to secure memory.

Workaround

Bypass the TZASC ID SWAP function by setting TZASC_ID_SWAP_BYPASS (bit 1) in the IOMUXC_GPR_GPR10 register.

ERR051126: USB3: Unreliable receiver detection in low power P3 mode at all temperatures

Description

The USB 3.0 controller enables the Receiver (Rx) Detection feature in low power mode 3 (P3 mode). However, USB 3.0 PHY does not reliably support receiver detection in P3 mode at all temperatures. Therefore, some USB 3.0 devices are not detected reliably in Super Speed mode or USB 3.0 PHY might falsely detect presence of a receiver when there is no device connected.

This erratum does not cause a compliance issue, because the receiver detection in P3 mode is beyond the PHY Interface For PCI Express, SATA, USB 3.1, DisplayPort, and Converged IO Architectures specification(PIPE) which only requires receiver detection in power mode 2 (P2 mode).

This erratum impacts Host USB3.0 PHY. It does not impact the USB3.0 PHY device operation.

Workaround

Set GUSB3PIPECTL[DisRxDetP3]=1 to configure USB 3.0 in P2 mode for receiver detection. (This workaround will be available in BSP from 2021 Q4)

ERR011324: USB3: USB device fails to connect after system resumes from DSM

Description

USB3 connection remains in inactive state after the system resumes from Deep Sleep Mode (DSM). USB3 module requires that 32K suspend_clk and AXI bus clock remain on for wake-up. However, relative clocks close in DSM. This conflict causes USB3 to remain inactive.

Workaround

After the system resumes, software can issue a USB link warm reset and cause USB3 to transition from U3 to U0 state.

ERR011231: USB: Clock must remain on during suspend/resume

Description

USB IP requires clock on during suspend/resume, to maintain RUN/STOP = 1.

This results in two impacts:

1. USB cannot be used as a wakeup source.
2. USB device mode will require extra power consumption as the clock is on.

Workaround

SW Workarounds:

Requires RUN/STOP bit to be set to 1 during suspend mode.

ERR011176: USB: USB 3.0 Host Hot Plug incorrectly enumerates

Description

The USB host hot plug causes the USB3 device to incorrectly enumerate.

Workaround

SW workarounds:

Two workarounds are possible to update the USB3.0 register settings within the Linux Driver:

Solution 1: Set GUSB2PHYCFG0.SUSPENDUSB20=0 when COMMONN=0

Solution 2: Set GUSB2PHYCFG.U2_FREECLK_EXISTS as 0

Solution 1 is the preferred workaround.

ERR011232: USDHC: uSDHC setting requirement for IPG_CLK and AHB_BUS clocks

Description

uSDHC AHB_BUS and IPG_CLK clocks must be synchronized.

Due to current physical design implementation, AHB_BUS and IPG_CLK must come from same clock source to maintain clock sync.

Workaround

Set AHB_BUS and IPG_CLK to clock source from PLL1.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6/2024
Document identifier: IMX8MDQLQ_2N14W